ASME Journal of Mechanical Design, Vol. 120, No. 1, pp 24-31.

Combining Interactive Exploration and Optimization for Assembly Design†

Gerard Jounghyun Kim
Department of Computer Science & Engineering
Pohang University of Science and Technology (POSTECH)
San 31, Hyoja-dong, Nam-gu
Pohang, Kyungbuk
Korea, 790-748
gkim@vision.postech.ac.kr

Simon Szykman
Engineering Design Laboratory
Manufacturing Systems Integration Division
National Institute of Standards and Technology
Gaithersburg, MD 20899
szykman@cme.nist.gov

ABSTRACT

This paper presents an integrated framework for conceptual assembly design. Because the complexity of assembly design leads to extremely large design spaces, adequate support of design space exploration is a key issue that must be addressed. CAMF allows the designer to manage the overall design process and explore the design space through explicit representation of design stages and their relationships (history), assembly design constraints, and rationale. The designer is free to use both bottom-up or top-down approaches to explore different assembly configurations. Exploration of the design space is further enabled by incorporating a simulated annealing-based refinement tool that allows the designer to rapidly complete partial designs, refine complete designs, and generate multiple design alternatives.

1 INTRODUCTION

In order to design and optimize a product, designers must be able to consider different alternatives, perform analysis to guide their own design process and focus in on a “good,” if not optimal, design. In practice, it is difficult to accomplish this using most current computer-aided design (CAD) systems because their implementations are geared toward supporting only a single level of design abstraction, that is, detailed geometry.

However, most engineering design processes proceed in series of stages, such as a functional design stage, a conceptual design stage, and a detailed design stage (Gui and Mantyla, 1994). During the functional design stage, functional requirements of a prospective product are identified. During the conceptual design stage, mappings from the required functional entities of the product to their physical forms occur. Preliminary shapes, sizes, orientations, materials, features, and locations of the physical forms may be determined. Then, the product design is refined to its final form during the detailed design stage through, for example, dimensioning, adding cosmetic features, surface modeling, etc.

One conventional method of handling exploration of design alternatives is through version control, whereby multiple versions of a design are stored as separate version- and/or date-stamped files as the design process proceeds. Version control of design files alone cannot sufficiently support design space

† An earlier version of this paper was awarded Best Paper, 1996 ASME Design Automation Conference, Irvine, CA, August.

exploration because each version of a design file simply represents a design snapshot in time. While changes from one version to the next may occur at different levels of abstraction, there is no explicit representation of these multiple levels. In contrast, the approach taken in this research is to organize and manage design alternatives according to different design stages of a given design process, which provides a much clearer picture of the expanding design space.

Kim (1995) introduced an assembly design tool called the Conceptual Assembly Modeling Framework (CAMF), which provides the ability to create and maintain evolving assembly designs, a mixture of top-down and bottom-up assembly modeling, and incorporation of tools such as analysis programs, design case bases and a materials library. CAMF does not prescribe the manner in which the designer approaches the assembly design, allowing a designer to move back and forth between levels within the representation (which can represent different views, design stages and/or levels of abstraction), incrementally modifying or augmenting various aspects of the current design.

This design strategy, which is commonly used by designers, is called the alternate use of abstraction and refinement (Paz-Soldan and Rinderle, 1989). As the designer moves back to earlier stages in the assembly process and changes previously-made design decisions, it is not uncommon for other existing parts of the design to become less practical or even rendered infeasible. Therefore, the refinement process is an integral part of the exploration of the design space. It is through refinement that these difficulties are resolved by either by applying local "patches" to the design, or by doing more substantial redesign at various design stages.

This paper presents advances that build on previous work to provide an interactive tool for assembly design, refinement and optimization with particular focus on providing support for design exploration for assembly configurations, and its impact on design-for-assembly (DFA) (Boothroyd and Dewhurst, 1989). This objective is achieved by incorporating a refinement tool, consisting of a simulated annealing optimization algorithm and a constraint solver, into the existing framework. These extensions to CAMF support interactive exploration by providing additional capabilities to the designer that are of use at various phases in the assembly design process. These include:

- rapid generation of multiple design alternatives (most useful at early phases),
- augmentation of partial design solutions through selection of values for certain types of variables (useful at various phases),
- generation of a feasible design given an infeasible design as a starting point (useful at various phases),
- refinement of designs created by the user (most useful at latter stages).

Because these capabilities are driven by optimization, exploration using the refinement tool is guided toward promising regions of the design space. In contrast to the majority of applications of optimization, the aim of this research is not to automate any part of the assembly design process, but rather to provide the human designer with a tool that is used interactively to improve search through a potentially very large design space. The design of a television remote control is used as an illustrative example to illustrate several of the capabilities of the refinement tool.

2 RELATED WORK: CAD FRAMEWORKS FOR ASSEMBLY MODELING

Relatively little attention has been paid to design or modeling methodologies for assemblies. Computer-aided design of assemblies can be categorized primarily into two groups, top-down (Gui and Mantyla, 1994; Popplestone, 1987; Rocheleau and Lee, 1987; Sodhi and Turner, 1991) and bottom-up approaches. Top-down assembly modeling is based on first generating a functional or symbolic description of a prospective design, and performing a stepwise refinement of component geometries. The functional model should be validated to some extent before moving into individual part design. A bottom-up approach starts with component design (along with a mental model of the design) and proceeds with continuous revision of the mental models and part design (Sodhi and Turner, 1994). In applying the alternate use of abstraction and refinement, designers mix both top-down and bottom-up approaches; thus, the CAMF framework has been created with the capability to support both approaches to assembly design.

To address the increasingly complex design activities in the context of concurrent engineering, van der Wolf (1984) describes the need for creating a new type of CAD framework and defines it as "a software infrastructure that provides a common operating environment for CAD tools". Key features of such CAD frameworks include an extended data management scheme for different types of design objects, a design process-driven user interface, and an open and flexible architecture – principles that are shared by CAMF. However, a majority of the work to date relating to CAD frameworks is oriented toward electronic circuit design (Brockman and Director, 1991; van der Wolf et al., 1990). Mechanical assembly design has different requirements in terms of user interaction and design exploration. Peplinski et al. (1995) describe a system for evaluating manufacturability at different abstraction levels and proposing a design based on the result.

3 THE CONCEPTUAL ASSEMBLY MODELING FRAMEWORK (CAMF)

3.1 Overview of CAMF

Within CAMF, a designer can specify both domain knowledge and problem-specific knowledge relating to an assembly design. The designer begins by creating a representation of the multiple stages in the design process, along with constraints that relate design decisions at various stages, and then proceeds to instantiate the design at these stages by adding additional detail. In the CAMF framework, the design process is viewed as sequence of state transitions where at each design state, design decisions are applied to achieve desired specifications, satisfy geometric constraints, and otherwise develop the design.

There are two different approaches toward explicitly communicating the design process to the designer. One is the often-used state-oriented representation, where the design process is represented by sequences of states of evolving design objects. A drawback to this approach is that it is difficult to graphically represent how design objects evolve into acceptable states during the design process. More specifically, because there may be a lack of geometric definition at the conceptual design stage, certain differences between successive design stages may not be visually apparent.

Another method is to use vocabularies of design decisions, or operators, that have been applied thus far to distinguish between different design states. This is called a decision-oriented representation of state-based transition for design. A brief explanation of the decision-oriented representation used in CAMF follows; further details regarding the representation can be found in (Kim, 1995). First, the term *design object* loosely refers to any object used, created, or modified during the design process in order to create the final design. Design objects include *function*, *specifications*, *constraints*, *parts*, *shapes*, etc. As the design process proceeds, the designer applies *design decisions* to create and manipulate these design objects.

In CAMF, different types of design decisions can be applied at different user-defined *design stages*. The user can define design stages at which only particular types of design decisions can be applied, producing a design step. For instance, in the assembly design process used for the remote control example in this paper, at a function-to-form mapping design stage the user may only apply operators related to creating new physical components for a given function. The design context consists of a set of design decisions applied at various design stages. CAMF allows the designer to maintain multiple design contexts (i.e. representations of multiple assembly designs) concurrently, from which different alternative designs can be generated.

CAMF uses the decision-oriented representation because it is more natural to describe and organize the assembly design process using design actions. Both symbolic and geometric descriptions of the evolving design are "derived" from the series of design actions when needed, rather than storing all the alternative designs in accumulation. This promotes reuse of certain design steps and also helps designers in focusing on certain design stages. Table 1 shows examples of different types of design objects and attributes that are represented in CAMF.

Figure 1 shows an overview of the architecture of CAMF. The work presented in this paper concerns the new refinement and optimization tool, shaded in gray in the figure. The central component of CAMF is the design process manager (DPM). The DPM allows the user to specify a generic design process model and then to design within that model in a structured manner. The DPM, which is described further in the next section, provides a graphical view of design evolution and alternatives using a

Table 1. Examples of design objects and some of their attributes.

Object	Attributes (partial list)
function	:function-of
subassembly	:has-function, :stability
part	:has-function, :size, :location
feature	:type, :location, :size
liaison	:type, :mating-parts
design rationale	:justifies, :type

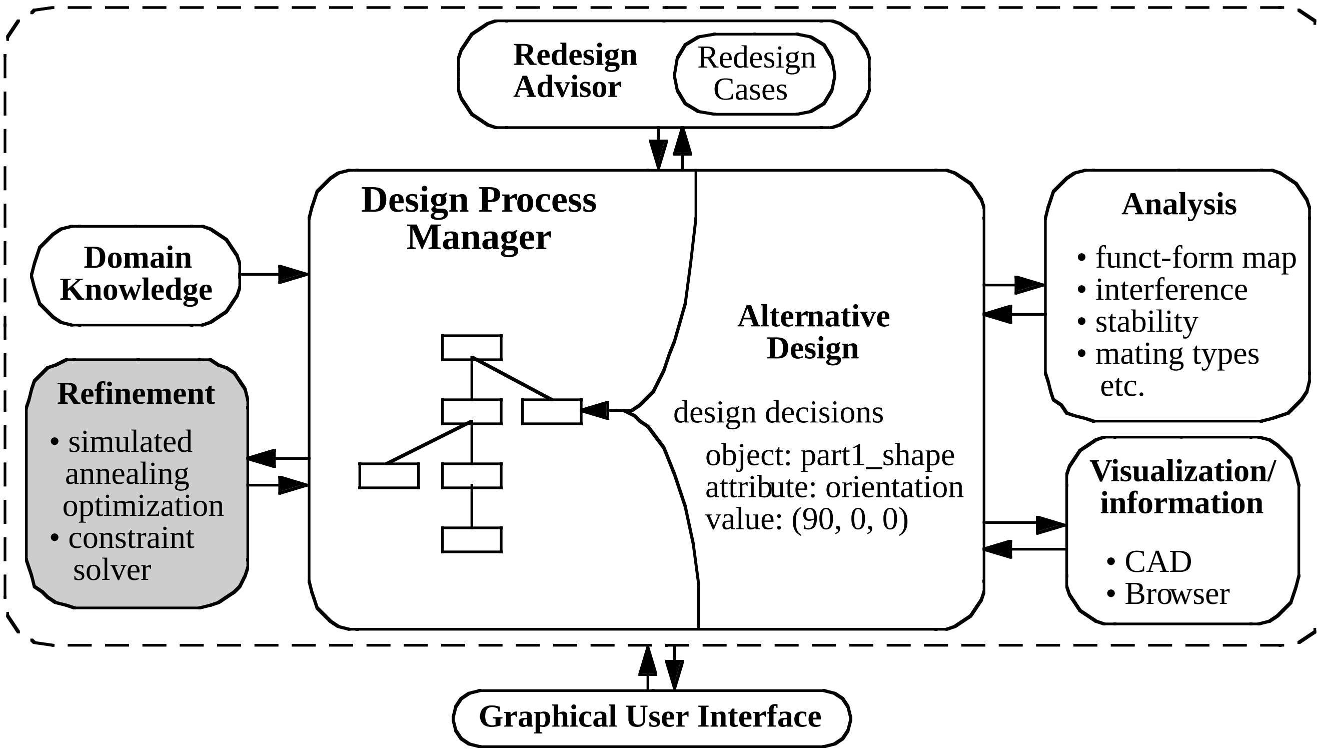

Figure 1. Architecture of the CAMF assembly design framework.

tree-like representation where each level corresponds to a particular design stage, each node corresponds to a design step, and branching indicates multiple design alternatives or contexts. This representation is illustrated in greater detail at the beginning of the following section.

The simulated annealing refinement and optimization tool is integrated with the DPM, which also includes several previously-incorporated design tools, namely, a geometric modeler (e.g. Spatial Technologies' ACIS Solid Modeler, SDRC's I-DEAS Master's Series[1]), a knowledge browser (for visualization), a number of design analysis tools (e.g. interference checking and evaluation of mating types), and a case-based redesign advisor (Kim and Bekey, 1994) that uses output from the analysis tool to generate new design alternatives.

CAMF currently does not ensure complete consistency nor validity of a design context; general knowledge about what is valid or feasible remains with the designer. However, the refinement tool (described in Section 4) is able to generate design alternatives that are consistent subject to given constraints on certain types of design attributes.

3.2 Assembly Design Process

Figure 2 illustrates a representation of a design space for the television remote control assembly design problem. The labels along the left column are design stages (specification selection, functional

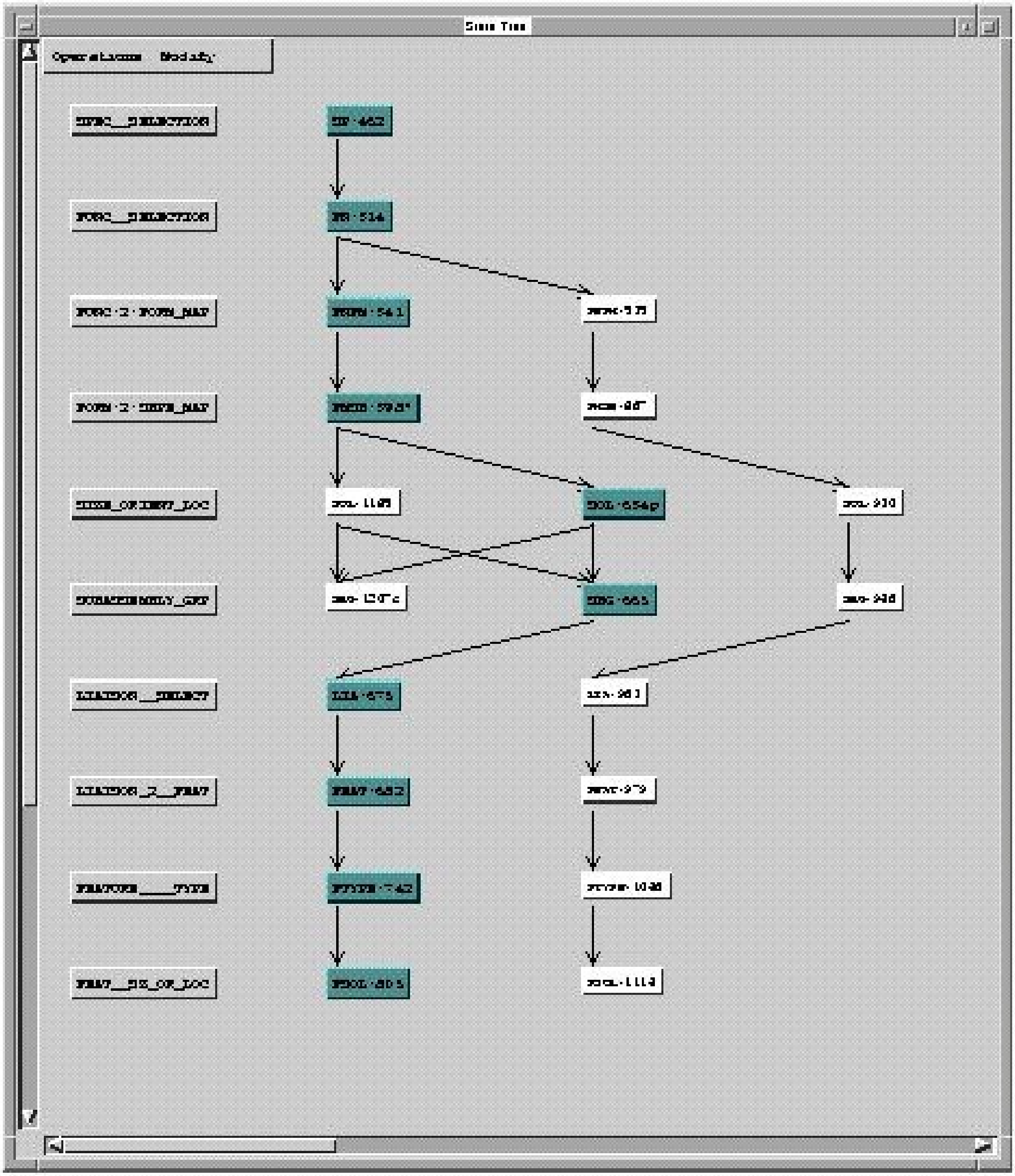

Figure 2. A design space for a television remote control assembly design task.

[1] Use of these commercial product names does not imply recommendation or endorsement by the National Institute of Standards and Technology.

Table 2. CAMF design stages for the television remote control example (see Figure 1).

Stage Name	**Mnemonic**	**Design Activity**
Specification Selection	SP	create and select set of specifications
Functional Element Selection	FN	create and select set of functional elements
Function-to-Form Mapping	FNFM	create conceptual forms and map FE's to conceptual forms
Form-to-Shape Mapping	FMSH	create and select approximate shapes for forms
Size, Orientation, and Location of Forms	SOL	select relative sizes, orientations and locations for shapes
Subassembly Grouping	SBG	define groupings of subassemblies
Liaison Selection	LIA	create liaisons among forms
Liaison-to-Feature Mapping	FEAT	create and select features for liaisons
Feature Type Selection	FTYPE	create and select shapes for features
Feature Size, Orientation and Location	FSOL	select relative sizes, orientations and locations for features

selection, etc.) defined for this problem by the user. Each of the nodes to the right represents a design step, where the node names are identifiers for a series of design decisions and supporting design rationale (not shown in the figure) associated with each node. A set of design steps connected by arrows represents a distinct design context, so that each path from a top node to a bottom node corresponds to a different design alternative. The designer can readily explore different alternatives by maintaining multiple solution paths and moving back and forth between them. In the figure, several paths through the design stages are possible; the highlighted one is the "current" design context which the designer is interacting with.

In CAMF, one may use a generic assembly design process model or define one's own for a specific design task. A new design process is created by defining new design stages and their order, and associating each of them with allowable design decisions and relevant constraints. This information is encoded in a file and imported to the system as domain knowledge. Table 2 lists the different design stages for the television remote control design example which were shown in Figure 2. Each design stage is associated with a particular set of objects and attributes that can be created or modified only during that stage. Such specification of design processes is input by the user to the system as domain or design knowledge. For example, during the function element selection stage, only functional elements can be created and they are associated with the assembly design through *:function-of* relation.

For the design process model shown in Table 2, design stages are related to different contributions to design costs. For instance, the function-to-form mapping stage determines the number of physical components in the assembly. The number of components is a very important assembly cost variable (because it has a direct relationship to required handling and feeding devices on the factory floor), and the designer might want to consider and maintain multiple alternatives for that stage. Different combinations of component sizes, orientations, and locations (see the fifth design stage, *SOL*, in Table 2) may result in a stable or unstable assembly, which has direct implications regarding fixture requirements.

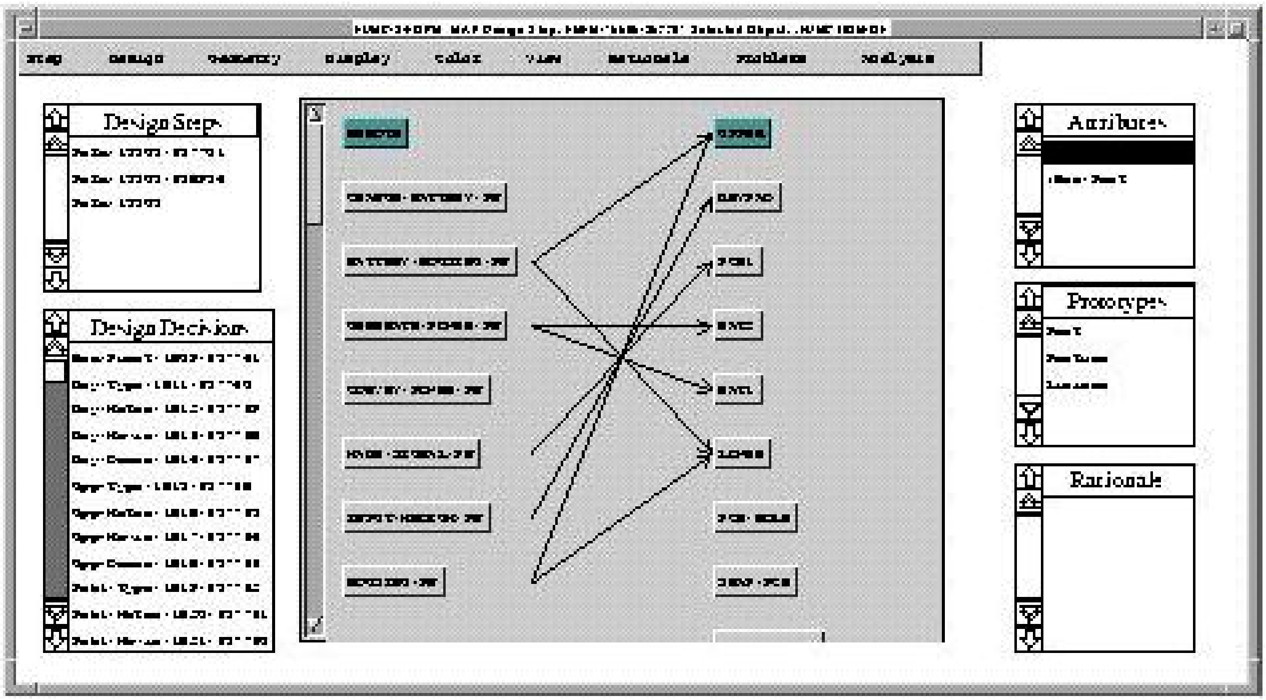

Figure 3. Design step, FNFM-535, in which *GENERATE-POWER-FN* is mapped to two physical components *BAT1* and *BAT2*.

4 DESIGN SPACE EXPLORATION

In this section, example interactive design sessions are presented where different assembly alternatives for the television remote control are created with CAMF. Section 4.1 illustrates manual design space exploration, where all design decisions are made by the designer. Due to lack of space, only a subset of the overall design process – selection of battery type and spatial configuration – is presented. Section 4.2 describes the simulated annealing algorithm, the approach which forms the basis for the optimization and refinement tool, and illustrates computer aided exploration at different stages of the design process.

4.1 Manual Design Space Exploration

Figure 2 showed a design space corresponding to the television remote control assembly. Initially, the design space starts out empty. During the first two design stages, specifications, functions, and their interrelationships are defined. At the third design stage (function-to-form mapping), two alternatives are generated by the designer: *FNFM-541* and *FNFM-535* (see the third row of Figure 2). The function-to-form mappings for the first alternative are illustrated in the knowledge browser interface shown in Figure 3. The second alternative is similar, except that the function *GENERATE-POWER-FN* is mapped onto a single part *9V-BAT*, instead of two parts *BAT1* and *BAT2*, as shown in the figure.

At the next design stage (form-to-shape mapping), the designer selects shapes for the two battery configurations. Design step *FMSH-598* corresponds to the selection of cylinders as generic shapes for the 2 AA batteries, while *FMSH-867* uses a rectangular block to approximate the shape of a 9 Volt battery. In the fifth design stage, different geometric configurations are explored using an interface to a geometric modeler and size, orientation and location operators. The three alternatives shown for the fifth stage in Figure 2 correspond to one in which two AA batteries are located end-to-end along the length of the remote, one in which they are seated side-by-side oriented with their length running the width of the remote control, and one in which the 9 Volt battery is oriented with its length along the width of the remote.

As further design refinements continue, one alternative among the three design alternatives is manually discarded by the user due to a preference for a design in which the batteries are seated side-by-side, which is less likely to interfere with the printed circuit board (PCB). Accordingly, that alternative ends at the sixth design stage and does not proceed to a “complete” solution at the bottom stage. At the appropriate stages for the remaining two alternatives, liaisons are selected, features are created and placed and two assembly designs are created.

At any design stage, the designer has the option of performing any applicable analysis (available through a menu seen in Figure 3) to help guide the design. Designers may focus on one part of the design, as is illustrated in this example, and then modify or complete unfinished parts of the design at preceding or subsequent design stages. The optimization-based refinement tool is available to the designer at all phases of the design process, providing the ability to generate multiple alternatives, augment partial designs, find feasible solutions given an infeasible design, and refine complete designs.

4.2 Computer Aided Design Exploration Using Simulated Annealing

4.2.1 Simulated Annealing. Simulated annealing is a stochastic optimization technique that was introduced by Kirkpatrick et al. (1983). At the start of an optimization using simulated annealing, the algorithm begins at an initial design state. The algorithm then takes a step to a new design state by perturbing the current design. The objective function value of the new state is compared to that of the previous state. If the new state is better than the previous one, it is accepted; if it is worse, it is accepted or rejected with some probability.

The probability of accepting an inferior state (i.e. a step in a direction away from an optimum) is a function of a parameter called temperature. Initially, the temperature (and therefore the probability of accepting inferior steps) starts out high. Since many inferior steps are accepted, this results in near-random exploration to find promising regions of the design space. As the optimization proceeds, the temperature decreases and fewer inferior steps are accepted, making the search less random. As the temperature continues to decrease, the algorithm reaches a point where the search resembles a downhill search because virtually no inferior steps are accepted. This allows the algorithm to converge to local optima in the current region of the design space.

The control of the temperature parameter is done using an annealing schedule, which is a critical part of a simulated annealing algorithm. The annealing schedule used for this research is an adaptive annealing schedule which, after the optimization begins, calculates an initial temperature according to a scheme proposed by White (1984) and calculates temperature reductions using a method described by Huang et al. (1986). The advantage of this approach is that the annealing schedule is tailored to a particular problem during the optimization, in contrast to a fixed annealing schedule where the schedule parameters are selected ahead of time and do not change during the optimization. The adaptive annealing schedule results in improved efficiency and convergence characteristics.

4.2.2 Formulation of the Optimization Problem. The previous section describes simulated annealing at a generic level. This section describes the problem-specific aspects of the algorithm formulation: the design variables, the method of perturbing designs and the objective function. In practice, fully automated search or design generation cannot be achieved at all design stages, nor over all design variables. This is because assembly design involves a great deal of knowledge which often cannot be encoded in a way that allows automated design decisions or computational evaluation of design alternatives. Variables for which sufficient knowledge is available to allow such automation are referred to as *refinement variables*.

For the television remote design example, the refinement variables are material for components, mating type for liaisons among parts, feature type for the matings and selection of generic shapes for those features. For this example, each of the refinement variables is changed at a different design stage, stages 3, 7, 8 and 9, respectively. However, this is not a general rule and it is possible to have more than one refinement variable at each design stage. It should also be noted that design stages may contain multiple types of variables. For instance, while material is the only refinement variable at the function-to-form mapping stage, the designer can make other decisions that affect various design attributes at that stage.

The input to the refinement phase is a design produced by the designer. This design may be an initial complete design or a partial design having values for one or more refinement variables left unassigned. At each iteration in the simulated annealing algorithm, a design stage that contains one or more refinement variables is randomly selected to be perturbed. Next, a design decision is chosen at random from that level, and one of the refinement variables is selected to be modified. The current value for that refinement variable is then changed to one selected from a list of feasible new values. This list is generated by taking the list of possible values for that variable, or computing the possible values from constraint knowledge, and removing the current value as well as any values that would violate constraints which have been previously entered as part of the domain knowledge.

Depending on the combination of values of refinement variables for other design decisions at various design stages, it is possible for the list of feasible new values to be empty because of constraints that affect allowable values (e.g. the mating type, or liaison, affects allowable materials and vice versa). When this occurs, a penalty is assigned to the objective function. As the optimization proceeds, these penalties are eliminated by resolving the source of the violation; that is, in a subsequent iteration, one of

Table 3. Interactive design using the simulated annealing refinement tool.

Design Stage	Object	Attribute	Refinement 1	Refinement 2	Refinement 3
Function-to-form mapping	upper	material	Steel	Wood	Plastic
	lower	material	Plastic	Wood	Plastic
	keypad	material	Steel	Rubber	Rubber
	pcb	material	Plastic	Plastic	Plastic
	battery1	material	Wood	Material-12	Material-12
	battery2	material	Wood	Material-12	Material-12
Liaison selection	liaison-upper-keypad	type	Solder	Inserting	Press-fit
	liaison-lower-pcb	type	Press-fit	Inserting	Press-fit
	liaison-battery-lower	type	Inserting	Inserting	Inserting
Feature selection	pcb-feature	type	Boss	Pad	Pad
	lower-feature	type	Groove	Slot	Slot
Feature shape selection	pcb-feature	has-shape	Circ-protrusion	Rect-protrusion	Rect-protrusion
	lower-feature	has-shape	Circ-hole	Rect-hole	Rect-protrusion

the constraints that caused variable values to be removed from the feasible list is rendered inactive by changing a different variable elsewhere. This results in a feasible list which is no longer empty. The next time that design decision is selected for a design perturbation, a feasible value can be found, eliminating the violation penalty.

4.2.3 Results. We now illustrate the use of the simulated annealing refinement tool as an aid for the exploration of the design space for the television remote control example. The designer begins by creating an initial design and using the refinement tool interactively, is able to rapidly guide design space exploration towards promising designs. The user interface provides ways to easily represent design constraints and specify the design attributes to be affected by the simulated annealing algorithm. The user inputs constraints relating refinement variables at different stages and specifies the design stages over which refinement is to take place. In this case, the designer selects all four stages that include refinement variables, though fewer could have been selected, and performs a series of interactive refinements.

Table 3 shows the results of applying the simulated annealing algorithm to the problem described above. (The table is not a comprehensive list of all design objects or attributes for this problem, but highlights those affected by the interactive exploration session). In the first run, the refinement was used to generate a design that improved the designer's initial attempt, subject to any relevant design constraints. In other words, the optimization-based refinement tool drove the design toward a solution having a lower assembly cost[2].

---

[2] Currently, a weighted sum of machining cost, assembly cost and material cost is used as a simplified measure of the overall cost.

Although there are constraints that relate refinement variables at various levels, these constraints do not fully describe needs for functionality or validity of the design due to the complexity of the assembly design task. Thus, we see that in the column for refinement 1, the refinement has resulted in the selection of wood as the material for the batteries. The designer realizes that he forgot to fix the battery material; because it was allowed to vary, the optimization selected a low cost material but one that is inappropriate for batteries. Via the user interface, the designer manually fixes the material to *Material-12*, which corresponds to the material cost of batteries in the CAMF materials library. The designer takes the opportunity to make a similar modification by making the keypad material rubber for human factor reasons.

The refinement algorithm is run a second time with these new constraints. The results are shown in Table 3, where shaded entries are variables that have changed from the previous refinement. Note that the changes in material have caused changes in liaisons (mating types), which have in turn affected the feature and feature shape selection at other design stages. After this refinement, the current design uses wood as the material for the upper and lower parts of the remote control housing (fifth column of Table 3) due to its low material cost. In practice, wood is not used despite a low material cost for a variety of reasons such as durability and marketability.

Rather than restricting design space exploration by constraining the housing material, the designer realizes that material cost is weighted too heavily relative to the other costs and adjusts the weights in the evaluation function, again through the user interface. The algorithm is run a third time and plastic materials are selected for the two housing components, again leading to changes in liaisons and feature shapes as shown in Table 3. Under the new weighting, more expensive materials were selected in order to allow less expensive mating types. This leads to lower assembly costs, and under the new weighting lower overall costs.

5 DISCUSSION

The assembly design tool developed through this work makes it easier for a designer to explore the design space associated with assembly design tasks. Through the use of this framework, a designer can represent a design process, and both represent and evaluate design artifacts. By using CAMF interactively, the designer can perform design refinement and redesign in a variety of ways – either by locally patching a design (extending a design at one design stage) or by moving back to previous levels of the hierarchy. At some stages, such as specification selection and function-to-form mapping, the exploration of the design space and the evaluation of designs is left to the user. At other stages, the designer can generate an initial attempt at a design, which may be partial or complete, and then call on a simulated annealing refinement tool for further interactive design space exploration.

The intent of this work is not to automate the design process, since developing a system that captures all relevant domain and design knowledge is in practice extremely difficult (if not impossible) for non-trivial domains. Rather, the aim is to create an interactive design tool that is able to aid the designer in generating alternatives, directing search towards good or optimal designs, and evaluating designs.

There are a number of motivations for the use of simulated annealing as a design space exploration technique for this work. Due to the combinatorial nature of the problem addressed in this research, an exhaustive search of the design space is infeasible. The size of the design space (where each design alternative corresponds to a unique path through the hierarchy of design stages) can become extremely large or even infinite depending on how the assembly design problem is structured. Simulated annealing, which has been applied to other classical combinatorial optimization problems (Collins et al., 1988) is able to find optimal or near-optimal designs without exhaustively enumerating of all possible alternatives, and is therefore well-suited for a problem such as this one.

A more traditional alternative to exhaustive enumeration would be to use a greedy search technique starting at the first design stage and moving through subsequent stages. The difficulty with this approach is that the ability to generate a given solution is dependent on the sequence that the design stages are visited, due to constraints between design attributes at different stages. For instance, consider a design problem where an optimal solution uses a particular liaison. If assignment of materials is done before assignment of liaisons, a decision that selects the best (e.g. lowest cost) material for a pair of parts may preclude the selection of the best liaison at a later stage because not all liaisons are possible with all materials. This would prevent a greedy search from finding the overall best solution.

The simulated annealing algorithm is able to avoid this difficulty due to the way in which the design refinement optimization is formulated; rather than refining the design sequentially by moving down the design stage hierarchy, at each iteration the algorithm can modify any of the applicable design stages. Making design decisions in this manner does not result in committing to a path that would make finding the global optimum impossible.

It should be noted that because the refinement variables are only a subset of the full set of design variables, the optimization is only exploring a subspace of the overall design space. Thus, while simulated annealing is commonly referred to as a global optimization technique, any least-cost designs produced by the optimization would be optimal with respect to the refinement variables which the optimization is able to modify. For this reason, the optimization is used interactively as an aid to designer exploration and not as a method for automating the assembly design process.

6 CONCLUSIONS

This paper presents an assembly design framework that creates and manages multiple design alternatives at different levels of abstraction according to a generic design process model in conjunction with domain-specific knowledge. Within the CAMF framework, design process management is achieved using a decision-oriented representation which allows the designer to represent knowledge about the design process and constraints, as well as information about the artifact being designed, design history, and design rationale.

Because the complexity of many assembly design problems can lead to extremely large design spaces, enabling adequate support of design space exploration is a key issue that must be addressed in a

CAD tool. CAMF enables both top-down and bottom-up approaches to assembly design by allowing the designer to freely move back and forth between design stages. The exploration of the design space is further supported by incorporating an interactive simulated annealing-based optimization tool that allows the designer to generate multiple design alternatives, rapidly complete partial designs, refine complete designs, and generate a feasible design given an infeasible design as a starting point.

The current framework provides the user with substantial flexibility to represent knowledge about the design process and artifact. One drawback to this approach is the difficulty that can arise with explicitly representing this knowledge within CAMF. While designs generated by the refinement/optimization tool are consistent subject to user-specified constraints that relate refinement variables to one another, knowledge about interactions between non-refinement variables resides largely with the designer. Thus, consistency of certain aspects of the design must be maintained by the user. Future work in expanding the scope of the refinement tool past the four currently-implemented design stages will require addressing these issues.

An area of future work that will affect the capabilities of CAMF is improvement of the design evaluation. Incorporation of more accurate DFM/DFA cost estimation tools, such as those developed as part of the SEER-DFM commercial software system (SEER, 1995), would be of benefit to this work. Because of the difficulty in automating the evaluation of some assembly-related attributes such as symmetry or tangling of parts, CAMF will remain an interactive design tool. However, as analysis and evaluation improve, search through the design space will be better guided toward good solutions, whether the exploration be manual or computer-aided using the refinement tool.

ACKNOWLEDGMENTS

This research was conducted at the National Institute of Standards and Technology and was supported by the National Research Council Postdoctoral Research Associateship program.

REFERENCES

Boothroyd, G. and P. Dewhurst (1989), *Product Design for Assembly Handbook*, Boothroyd and Dewhurst, Inc.

Brockman, J. and S. Director (1991), "The Hercules CAD Task Management System," *Proceedings of IEEE ICCAD-91*, pp 254-257.

Collins, N., R. Eglese, and B. Golden (1988), "Simulated Annealing – An Annotated Bibliography," *American Journal of Mathematical and Management Sciences*, **8**(3&4):209-307

Gui, J. and M. Mantyla (1994), "Functional Understanding of Assembly Modeling," *Computer-Aided Design*, **26**:435-451.

Huang, M., F. Romeo, and A. Sangiovanni-Vincentelli (1986), "An Efficient General Cooling Schedule for Simulated Annealing," *ICCAD-86: IEEE International Conference on Computer-Aided Design – Digest of Technical Papers*, Santa Clara, CA, November, pp 381-384.

Kim, G. J. (1995), “A Decision-Based Framework for Exploring Assembly Configuration,” *CE95: Proceedings of the Second International Conference on Concurrent Engineering, Research and Applications*, McLean, VA, August.

Kim, G. J. and G. Bekey (1994) “Design-for-Assembly by Reverse Engineering,” *Artificial Intelligence in Design ‘94*, J. S. Gero, ed., Kluwer Academic Publishers, pp 717-734.

Kirkpatrick, S., C. Gelatt, and M. Vecchi (1983), “Optimization by Simulated Annealing,” *Science*, **220**(4598):671-679.

Paz-Soldan, J. and J. R. Rinderle (1989), “The Alternate Use of Abstraction and Refinement in Conceptual Mechanical Design,” *Proceedings of the 1989 ASME Winter Annual Meeting*, San Francisco, CA, December.

Peplinski, J., P. Koch, J. Allen, and F. Mistree (1995), “Flame: A Manufacturability Evaluator for Use at Different Levels of Abstraction,” *Proceedings of 1995 ASME Design Engineering Technical Conferences*, Boston, MA, September, pp 893-900.

Popplestone, R. (1987), “Edinburgh Designer System as a Framework for Robotics,” *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, pp 1972-1977.

Rocheleau, D. and K. Lee. (1987), “System for Interactive Assembly Modeling,” *Computer-Aided Design*, **19**:65-72.

SEER (1995), *SEER-DFM Users Guide*, G A SEER Technologies, El Segundo, CA.

Sodhi, R. and J. Turner (1991), “Representing Tolerance and Assembly Information in a Feature Based Design Environment,” *Proceedings of the 1991 ASME Design Automation Conference*, Miami, FL, September, pp 101-108.

Sodhi, R. and J. Turner (1994), “Towards Modeling of Assemblies for Product Design,” *Computer-Aided Design*, **26**:85-97.

van der Wolf, P. (1994), *CAD Frameworks: Principles and Architecture,* Kluwer Academic Publishers.

van der Wolf, P., G. W. Sloof, P. Bingley, and P. Dewilde (1990), “On the Architecture of a CAD Framework: The Nelsis Approach,” *Proceedings of European Design Automation Conference*, pp 29-33.

White, S. (1984), “Concepts of Scale in Simulated Annealing,” *Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers*, Port Chester, NY, October, pp 646-651.